Histograms Introduction

In digital image processing, the histogram is used for graphical representation of a digital image. A graph is a plot by the number of pixels for each tonal value. Nowadays, image histogram is present in digital cameras. Photographers use them to see the distribution of tones captured.

In a graph, the horizontal axis of the graph is used to represent tonal variations whereas the vertical axis is used to represent the number of pixels in that particular pixel. Black and dark areas are represented in the left side of the horizontal axis, medium grey color is represented in the middle, and the vertical axis represents the size of the area.

Histograms - 1 : Find, Plot, Analyze

Find histograms, using both OpenCV and Numpy functions

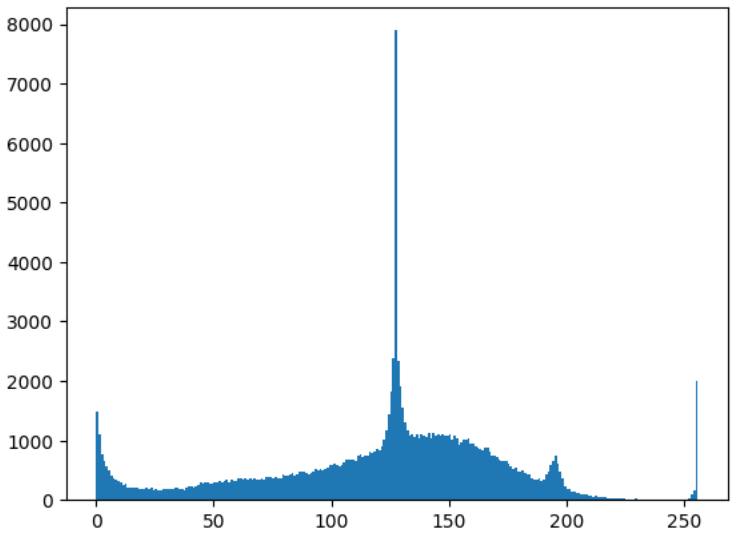
Plot histograms, using OpenCV and Matplotlib functions

We will see these functions : cv.calcHist(), np.histogram() etc.

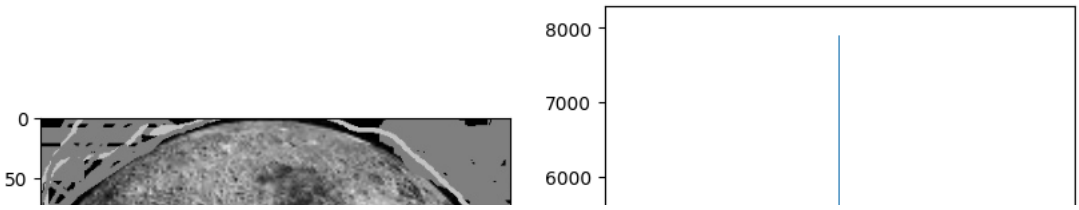
what is histogram ? You can consider histogram as a graph or plot, which gives you an overall idea about the intensity distribution of an image. It is a plot with pixel values (ranging from 0 to 255, not always) in X-axis and corresponding number of pixels in the image on Y-axis.

By looking at the histogram of an image, you get intuition about contrast, brightness, intensity distribution etc of that image. Almost all image processing tools today, provides features on histogram.

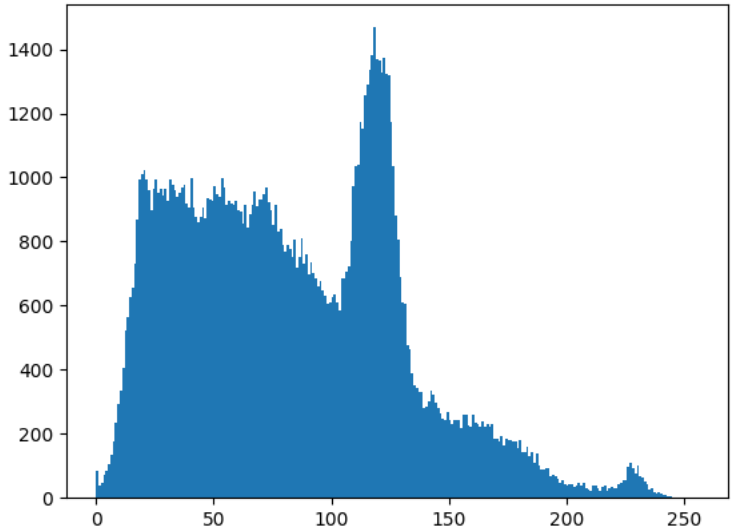
```
import cv2 as cv
import matplotlib.pyplot as plt
img = cv.imread('Images/moon1.jpeg', cv.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"
hist = cv.calcHist([img],[0],None,[256],[0,256])

plt.hist(img.ravel(),256,[0,256]); plt.show()
```



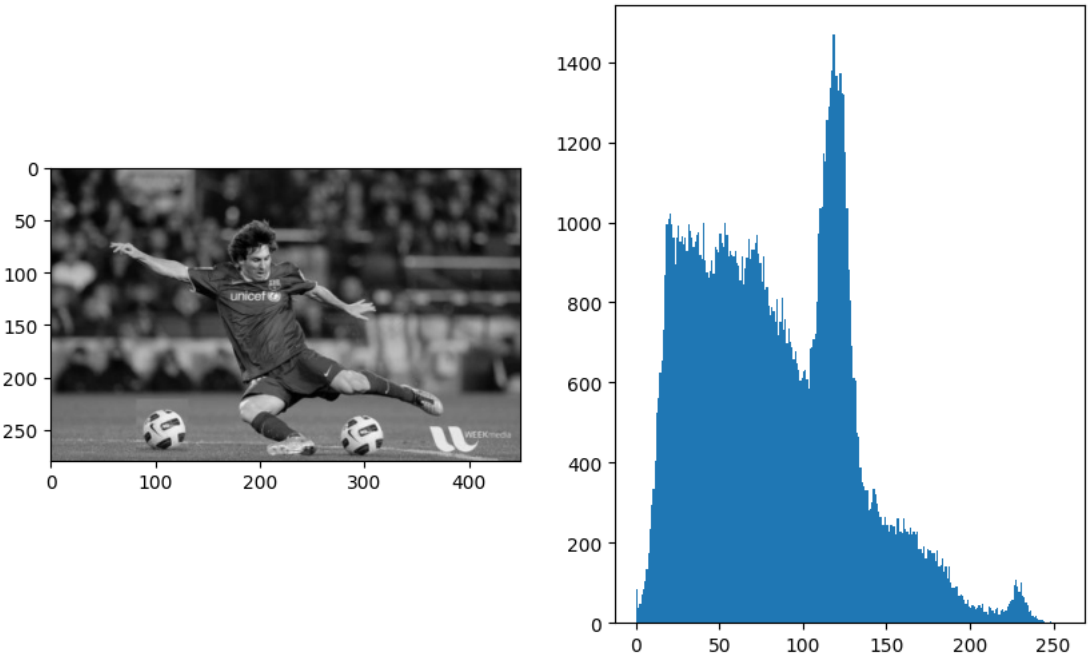

```
plt.figure(figsize=(10,6))
plt.subplot(121), plt.imshow(img, cmap="gray")
plt.subplot(122), plt.hist(img.ravel(),256,[0,256]); plt.show()
```



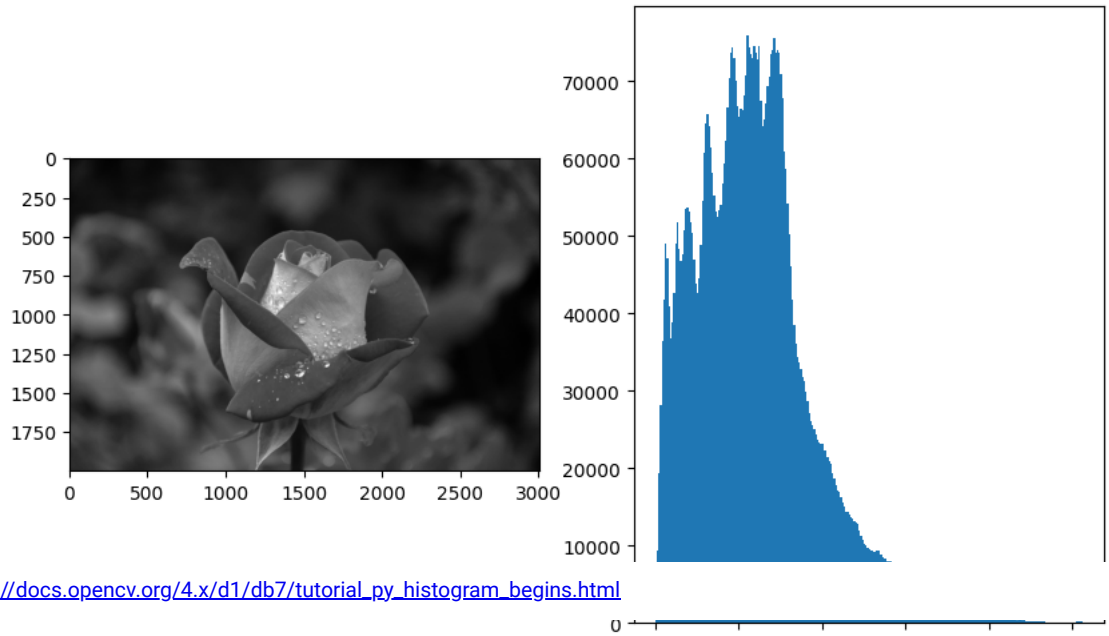
```
import cv2 as cv
import matplotlib.pyplot as plt
img = cv.imread('Images/messi5.jpg', cv.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"
plt.hist(img.ravel(),256,[0,256]); plt.show()
```



```
plt.figure(figsize=(10,6))
plt.subplot(121), plt.imshow(img, cmap="gray")
plt.subplot(122), plt.hist(img.ravel(),256,[0,256]); plt.show()
```

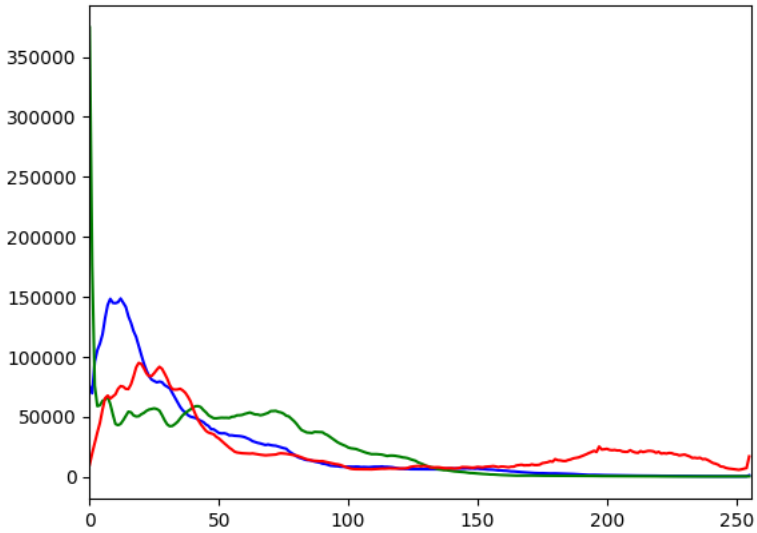


```
img = cv.imread('Images/rose1.jpg', cv.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"
#hist = cv.calcHist([img],[0],None,[256],[0,256])
plt.figure(figsize=(10,6))
plt.subplot(121), plt.imshow(img, cmap="gray")
plt.subplot(122), plt.hist(img.ravel(),256,[0,256]); plt.show()
```



https://docs.opencv.org/4.x/d1/db7/tutorial_py_histogram_begins.html

```
img = cv.imread('Images/rose1.jpg')
color = ('b','g','r')
for i,col in enumerate(color):
    histr = cv.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.show()
```



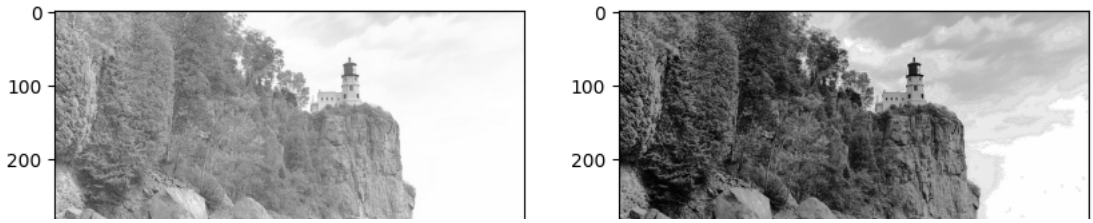
Histograms - 2: Histogram Equalization

Consider an image whose pixel values are confined to some specific range of values only. For eg, brighter image will have all pixels confined to high values. But a good image will have pixels from all regions of the image. So you need to stretch this histogram to either ends (as given in below image, from wikipedia) and that is what Histogram Equalization does (in simple words). This normally improves the contrast of the image.

![Screenshot 2023-10-19 223906.png](<attachment:Screenshot 2023-10-19 223906.png>)

```
img = cv.imread('Images/hill.jpg', cv.IMREAD_GRAYSCALE)
assert img is not None, "file could not be read, check with os.path.exists()"
equ = cv.equalizeHist(img)
plt.figure(figsize=(10,5))
plt.subplot(121), plt.imshow(img, cmap="gray")
plt.subplot(122),plt.imshow(equ, cmap="gray")
```

```
(<Axes: >, <matplotlib.image.AxesImage at 0x18174825690>)
```



```
#https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html
# create a CLAHE object (Arguments are optional).
clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
cl1 = clahe.apply(img)
plt.imshow(cl1,cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x18174e96990>
```